AD-A067 947 CALIFORNIA UNIV LOS ANGELES GRADUATE SCHOOL OF MANAGEMENT F/G 9/2
ON THE USE OF PRODUCTIVITY AIDS IN SYSTEM DEVELOPMENT AND MAINT--ETC(U)
JAN 79 B P LIENTZ, E B SWANSON N00014-75-C-0266
UNCLASSIFIED TR-79-1 NL

1 OF 1
AD
A067947

END
DATE
FILMED
6-79
DDC

6 ON THE USE OF PRODUCTIVITY AIDS
IN SYSTEM DEVELOPMENT AND MAINTENANCE

11 15 Jan 79

14 TR-79-1

12 21p.

9 Technical rept.

10 Bennet P. Lientz [redacted] E. Burton Swanson
Graduate School of Management
University of California, Los Angeles*

Abstract

There exists a substantial literature on productivity aids. Software aids have been developed in areas such as testing, documentation, programming, design, and analysis. A survey of almost five hundred organizations was performed to analyze various aspects of software maintenance. A purpose of the survey was to analyze the use of various productivity aids, and to ascertain relationships between their use and maintenance effort and characteristics. The survey indicates that no productivity aid was widely employed in system development and that many productivity aids do not address perceived problem areas. Areas perceived as problems include user involvement in the application, handling user enhancements, and management concerns with resource allocation. Statistical analysis found certain tools (HIPO, automated flowcharting, data base dictionaries, and test data generators) → (cont on p 2)

---

*This work was partially supported by the Information Systems Program, Office of Naval Research under contract N00014-75-C-0266, project number NR 049-345.

DISCLAIMER NOTICE

(cont)

more prevalent among organizations with larger data processing budgets. Use of HIPO (automated flowcharts) was found to be more prevalent among younger (older) applications. Data base dictionaries and test data generators were found to associate with larger systems. Analysis revealed that the impact of the use of tools on maintenance effort is dominated by characteristics of size, number of reports, system development experience, and use of data base management systems.

1. INTRODUCTION

The use of productivity aids has been widely advocated as a means of improving software quality, reducing software maintenance effort, and in some cases, reducing development costs in some phases of design and implementation. A variety of tools and techniques have been proposed for different activities, including analysis (e.g., structured analysis, DBDA), design (e.g., structured design, ISDOS-PSL/PSA, top down design), programming (e.g., structured programming), documentation (e.g., HIPO, automated flowcharting), management and review (e.g., structured walkthrough, chief programmer team), and testing (e.g., test data generators). Less consideration has been given to impact assessment of such tools and techniques on operational systems. Some questions that arise are:

- How widely are such tools employed?

- Does the use of tools reduce maintenance effort?

Although generalizations must be made with caution (due to organization differences, differences in systems, etc.), it is necessary to study such questions to determine where improvements are needed and where tools may be helpful.

The preceding discussion of tools and techniques is part of the larger problem of understanding development and maintenance. Maintenance here is defined as any work on a current, operational application system. Although a great deal of effort has been directed at development, software maintenance has received less attention. A review of the literature and issues in software maintenance of application systems appears in Lientz and Swanson (2). Based on the need for

analyzing maintenance, a preliminary summary of eighty-six organizations was undertaken. The survey was based on a questionnaire in managerial, as well as technical, aspects of maintenance. The results of the survey are given in Lientz, Swanson, and Tompkins [4]. We will refer to this study later as the initial survey. Because of the interest generated by the survey and because of the need for a larger sample to consider specific issues, a larger scale survey was conducted. The questionnaire was reviewed and revised. It was divided in two parts--Part I dealing with the organization and Part II dealing with a specific application. The Data Processing Management Association (DPMA) agreed to support the survey by providing a randomly generated mailing list of 2000 data processing managers and supervisors as well as an enclosing cover letter to the questionnaire. Responses to the questionnaire were received from 487 organizations. This response rate (approximately 25%) is considered excellent considering the length and complexity of the questionnaire and the factor that limited resources permitted only a post-card follow-up. The responses were edited, keypunched, and verified. Statistical analysis was performed using the SPSS Statistical package. Some tabular results of questions were presented at the Annual DPMA meeting of November, 1978 (see Lientz and Swanson [3]).

In this paper we address the use of some productivitiy tools and techniques in reference to maintenance. We consider not only the use of such tools, but the environment in which the tools are employed. The environment includes organization size, system age, system size, and perceived problem areas in maintenance. A more comprehensive analysis of the survey is currently being prepared in monograph form. It should be emphasized that no judgement is being applied as to whether the extent of maintenance is good or bad. This obviously depends on the particular setting for an application.

2. THE MAINTENANCE ENVIRONMENT

Maintenance and operations are the part of the life cycle where many of the benefits of productivity tools are claimed to lie. The maintenance environment includes the perceived importance of specific tools and techniques as well as their effectiveness. Related to this are the problem areas perceived by management, as well as the measurement data collected during maintenance. More technical parts of the environment include the sources of enhancement and maintenance requests, the size and growth of the application system, and the age of the system.

The environment can affect the selection and use of specific tools and techniques For example, if data is not collected on maintenance effort, it may be difficult to justify productivity aids since the benefits could not easily be quantified. If design problems are perceived to be severe or documentation is faulty, then these areas might receive more attention for potential improvement.

o Organization Environment

In the survey we asked whether application programmers and/or systems analysts doing maintenance were organized separately from those doing system development. A separate maintenance organization would facilitate measurement and evaluation of tools. The response of the sample indicated that only 16.2% (79 out of 487) organized maintenance separately. We see from this that the measurement problem is more complex in the vast majority of the organizations, because of a lack of separation.

If productivity tools are to affect maintenance, then we need to know how much of the total systems effort is involved in maintenance. The survey indicated that the percentage in maintenance has remained at about 50% (to be precise, 48.5%) over the most recent two-year period. We also asked for the percentage of application programmers and systems analysts relative to the systems organization. The average was approximately 40%. Note that one could then estimate average estimated improvement of a technique which claimed to reduce maintenance by X%. Suppose technique Z claimed to reduce maintenance by 20%. The net savings to the organization would be 4% (.2 x .4 x .50) of the total systems organization. Of course, this is fictitious and only intended to show how the impact could be derived. Each organization would have to be addressed separately for a specific method. The point is clear, however. The impact on savings is limited by the numbers of programmers and systems analysts and by the effort in maintenance.

o Management Controls and Measurement

The survey asked which of nine management controls were employed in maintenance. The results appear in Figure 1. Productivity aids are sometimes claimed to reduce the number and extent of changes to the system. The figure reveals that, for many organizations, measurement of the effectiveness of tools would be difficult due to the absence of accounting controls. For example, some methods claim to reduce testing effort by insuring system simplicity. However, more than 40% of the organizations do not have a formal re-test procedure. In such cases, measurement of testing effort would be difficult and would rely on informal testing. This is particularly likely to be the case in smaller organizations, which cannot easily absorb the overhead associated with the use of formal controls.

Figure 1: <u>Use of Management Controls</u>

		Absolute Frequency	(Percent)
a.	All user requests for changes to the application system must be logged and documented.	384	(78.9)
b.	All user requests for changes to the application system must be cost justified.	160	(32.9)
c.	All troubles encountered in the operational processing of the application system programs must be logged and documented.	250	(51.3)
d.	All changes to the application programs must be logged and documented.	375	(77.0)
e.	All changes to the application programs must undergo a formal re-test procedure.	285	(58.5)
f.	With the exception of emergency fixes, all changes to the application programs are batched for periodic implementation according to a predetermined schedule.	137	(28.1)
g.	A formal audit of the application system is made periodically.	158	(32.4)
h.	Equipment costs associated with operating and maintaining the application system are charged back (in whole or in part) to the user.	163	(33.5)
i.	Personnel costs associated with operating and maintaining the application system are charged back (in whole or in part) to the user.	150	(30.8)

o Maintenance and Enhancement

Since specific tools address individual system activities, it is of interest to note where time is expended in maintenance and enhancement. This is shown in Figures 2 and 3, respectively. Allocation categories are based on those developed in Swanson [5]. Figure 2 reveals that about 40% of the effort is expended in user enhancements--providing new features or capabilities. Only 12% goes to fixing programs on an emergency basis. The effort to improve documentation and improve system efficiency together consume less than 10% of the maintenance resources. Figure 3 provides a breakdown of the 40% of the total effort in maintenance devoted to user enhancements. Almost two-thirds of the enhancement effort is seen to consist of giving the user "more." The rates here indicate that methods addressing reporting flexibility would appear to be of substantial benefit in maintenance. We will return to this when we consider problem areas perceived by management.

o Technical Characteristics of Applications

The applications studied in the survey were those which involved substantial maintenance effort, were significant to the organization, and which had been operational for at least one year. The breakdown of applications in the survey is given in Lientz and Swanson [3] Of interest are the size and other characteristics of the applications. The average age of applications was four to five years. The average number of programs in the application currently and one year ago was respectively 126.5 and 115.6. This indicates a growth of about 10% per year. The average number of source language statements

Figure 2: Allocation of Effort in Maintenance

		Mean Percent
a.	Emergency program fixes	12.0
b.	Routine debugging	9.0
c.	Accommodation of changes to data inputs and files	16.8
d.	Accommodation of changes to hardware and system software	6.0
e.	Enhancements for users	40.3
f.	Improvement of program documentation	5.3
g.	Recoding for efficiency in computation	3.9
	Others	3.3

Figure 3: Allocation of Effort in User Enhancements

		Mean Percent
a.	Providing new, additional reports	38.0
b.	Adding data to existing reports	25.3
c.	Reformating existing reports, without changing their data content	9.3
d.	Condensation of data in existing reports	5.2
e.	Consolidation of existing reports, reducing the number of reports	6.0
	Others	9.4

increased from 43.6K to 48K--again an increase of about 10%. The size of the data base or files increased by 5% from 41.6 M bytes to 43.8 M bytes. The number of pre-defined user reports increased from an average of 49.0 to 53.8, approximately 10%. The source languages used were dominated, as expected, by COBOL (51.6%). RPG was second at 22.4%, followed by Assembler at 11.9%.

The composite picture here is that the dominant language remains COBOL. Application systems, measured in a variety of ways, are growing at a rate of about 10% per year. Again, we must caution that these figures are based on the systems selected by respondents. In terms of hardware, the 10% growth rate can easily be accommodated by increased capacity due to hardware performance-price ratio improvements.

o *Perceived Problem Areas*

Respondents were asked to indicate the severity of various problems on a scale of 1 to 5, interpreted as follows: 1 - no problem at all; 2 - somewhat minor problem; 3 - minor problem; 4 - somewhat major problem; 5 - major problem.

The rankings based on medians are as follows:

Factors

Minor problem:

- o user demands for enhancements and extensions to the system*
- o competing demands for maintenance programming personnel time**
- o inadequate training of user personnel*
- o turnover in user organization*
- o meeting schedule commitments**
- o quality of application system documentation[+]

Somewhat minor problem:

- o motivation of maintenance personnel
- o forecasting of maintenance programming requirements
- o maintenance programming personnel
- o unrealistic user expectations
- o adherence to programming standards
- o adequacy of systems design specification
- o data integrity in application system
- o processing time requirements
- o turnover of maintenance personnel
- o changes made to system hardware and software
- o skills of maintenance programming personnel
- o quality of original programming
- o number of maintenance personnel available
- o lack of user understanding of application systems

No problem:

- o lack of user interest
- o system hardware and software reliability
- o storage requirements of application system
- o budgetary pressures

Reviewing these we see that most technical issues are no more than Somewhat Minor. In the Minor Problem category we have labeled issues of a user orientation by an asterisk (*), a management concern by a double asterisk (**), and a technical concern by a plus (+). Only one technical issue appears--quality of original documentation. Although turnover of maintenance personnel is not cited among the top problems, elsewhere in the survey it was found that only about half of the maintenance personnel were involved in the development of the system. This explains in part the need for good original documentation of the system.

Of the six Minor Problem concerns, the user organization is involved in three. User demands for enhancements has already been mentioned. Of the other two, one is potentially a management issue--turnover in user organizations. The other--user training--could be addressed by applying new or existing techniques.

The two management problems perceived as Minor--competing demands on personnel time and meeting scheduled commitments--reflect a need for measurement, estimation, and control. Competing demands on personnel time may reflect the gap between limited resources and user needs. The problem of meeting scheduled commitments may, in part, be due to a lack of controls and data from which to estimate effort.

3. USE OF TOOLS AND RELATIONSHIPS WITH OTHER FACTORS

o <u>Use of Selected Productivity Aids</u>

In the initial survey the sample was asked as to the tools employed in the maintenance of the system. Based upon these findings, a list of tools

Figure 4: Use of Design and Programming Aids In Systems Maintainance

	Tool	Frequency of Use (Percent)
a.	Decision tables	46.4
b.	Test data generator	36.2
c.	Chief programmer team	30.4
d.	Online programming	30.4
e.	Database dictionary	26.1
f.	Structured programming	24.6
g.	Structured walk-through	17.4
h.	Automatic flowcharting	10.1
i.	HIPO	7.2
j.	ISDOS (Automated design aid)	4.3

Figure 5: Use of Design and Programming Aids In Systems Development

	Tool	Absolute Frequency	(Percent)
a.	Decision tables	164	(33.7)
b.	Data base dictionary	73	(15.0)
c.	Test data generators	80	(16.4)
d.	Structured programming	145	(29.8)
e.	Automated flowcharting	24	(4.9)
f.	HIPO (Hierarchy plus Input-Process-Output) Design Aid Technique	34	(7.0)
g.	Structured walk-through	82	(16.8)
h.	Chief programmer team	187	(38.4)
	Others	47	(9.6)

was provided in the expanded survey. Respondents were asked to indicate which of these were employed in the development of the system. The results, shown in Figure 4, indicate the lack of tools in use. However, from the previous section we see further that many of these tools do not address the areas of concern expressed. Of the tools in Figure 5, three relate particularly to documentation (HIPO, decision tables, and automated flowcharting). However, none deal with the user and management problems cited.

o <u>Use of Tools and Size of Data Processing Budget</u>

The use of tools was analyzed in relation to the size of the data processing installation using cross tabulation and the chi square test. Larger organizations tend to use test data generators, data base dictionary, automatic flowcharting, and HIPO (significance levels--all less than .04%). This is to be expected since some of these tools require a substantial hardware environment--more likely to be present in organizations with larger data processing budgets. No relationships were found for decision tables, structured programming, or structured walk-throughs. Use of a chief programmer team was found to be more prevalent for organizations with medium range data processing budgets ($250,000 to $200,000 per year; significance level--less than 1%).

o <u>Use of Tools and System Age and Size</u>

Analysis of variance was employed to relate system age to the use of productivity aids in development. The use of HIPO was significantly associated

with younger systems at the 5.5% level. The use of automated flowcharting, on the other hand, was found to be associated with older systems (at the 10.8% level). There was a limited tendency for the use of data base dictionaries, structured programming, and structured walk-through with younger systems, but this was not of notable statistical significance. Decision tables, test data generators, and chief programmer teams did not associate with age.

Similar statistical analysis was performed for the number of programs in the system. Significantly associated with larger number of programs were data base dictionaries (9.1% level), and test data generators (5.8% level). Analysis of variance for the rate of growth in the number of programs found for increasing growth rates--data base dictionaries (0.8%) and chief programmer teams (5.1%).

o Use of Tools and Maintenance Effort

It is of obvious interest to determine if there is any relation between the use of tools and the amount of the maintenance effort. A preliminary effort suggested variable transformations to meet normality assumptions in the statistical model. Natural logarithm transforms were used for effort and size parameters. Stepwise linear regression was performed with the dependent variable--the natural logarithm of maintenance effort. The variables entered in order were: 1 - natural logarithm of the number of programs in the system currently, 2 - natural logarithm of the number of source language statements, 3 - natural logarithm of the size of the data base, 4 - system development experience, 5 - use of a data base management system, 6 - natural logarithm of the number of reports generated by the system currently.

The seventh variable entered is the first evidence of a tool--automated flowcharting. Overall then, the analysis indicates that the tools have a limited relationship with maintenance effort. Using a cut-off level of F=1.0, six of the eight tools eventually entered the regression equation. Even though the coefficients of the tool variables are very low, it is interesting to note that some may, in certain cases, appear actually to contribute to increased maintenance effort. The variables with positive relations to effort are automated flowcharting, data base dictionary, structured programming, and structured walk-through. Tools and techniques which are negatively related to maintenance effort were decision tables and chief programmer teams.

A similar analysis was performed with the dependent variable being the number of persons maintaining the system. Here the only tool negatively related to the number of people in maintenance of the particular application was the chief programmer team. Caution in interpreting these results is needed due to the number of smaller organizations with very limited maintenance staffing.

4. CONCLUSIONS

We have presented some of the results of a maintenance study from the perspective of the use of software productivity aids. The findings are limited by the extent of the survey. It should be emphasized that the survey was limited to standard business applications rather than aerospace or engineering applications. The results of the survey are that many productivity aids are not widely used. Second, the areas addressed by many methods miss areas perceived as problems by respondents. Third, there are areas that deserve more attention--specifically

techniques that enhance the users' role, and methods dealing with management concerns. An extension of the findings has been the development of a long-range information services planning methodology which has the control of maintenance and enhancement effort as one of its goals (See Lientz and Chen [1]).

References

1. Lientz, Bennet P. and Chen, M. "A Methodology for Long Range Information Services Planning." Long Range Planning (to appear July 1979).

2. Lientz, Bennet P. and Swanson, E. Burton. "Discovering Issues in Software Maintenance." Data Management, October 1978, pp. 15-18.

3. Lientz, Bennet P. and Swanson, E. Burton. "A Survey of Application Software Maintenance." An invited paper presented at DPMA 1978 Annual Meeting, November 1978.

4. Lientz, Bennet P.; Swanson, E. Burton; and Tompkins, G.E. "Characteristics of Application Software Maintenance." Comm. ACM, June 1978, pp. 466-471.

5. Swanson, E. Burton. "The Dimensions of Maintenance." Proceedings of the Second International Conference on Software Engineering. San Francisco: October 13-15, 1976.

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical 79-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) On the Use of Productivity Aids in System Development and Maintenance		5. TYPE OF REPORT & PERIOD COVERED Technical report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bennet P. Lientz and E. Burton Swanson		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0266
9. PERFORMING ORGANIZATION NAME AND ADDRESS Graduate School of Management University of California Los Angeles, CA. 90024		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 049-345
11. CONTROLLING OFFICE NAME AND ADDRESS Information Systems Program Office of Naval Research, Arlington, Virginia		12. REPORT DATE Jan. 15, 1979
		13. NUMBER OF PAGES 20
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

distribution of this document is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

productivity aids
system development
maintenance
maintenance survey

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

There exists a substantial literature on productivity aids. Software aids have been developed in areas such as testing, documentation, programming, design, and analysis. A survey of almost five hundred organizations was performed to analyze various aspects of software maintenance. A purpose of the survey was to analyze the use of various productivity aids, and to ascertain relationships between their use and maintenance effort and characteristics. The survey indicates that no productivity aid was widely employed in system development and that many productivity aids do not address perceived problem areas. Areas perceived as problems include

(over)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (contd.)

user involvement in the application, handling user enhancements, and management concerns with resource allocation. Statistical analysis found certain tools (HIPO, automated flowcharting, data base dictionaries, and test data generators) more prevalent among organizations with larger data processing budgets. Use of HIPO (automated flowcharts) was found to be more prevalent among younger (older) applications. Data base dictionaries and test data generators were found to associate with larger systems. Analysis revealed that the impact of the use of tools on maintenance effort is dominated by characteristics of size, number of reports, system development experience, and use of data base management systems.